

# 系统容量优化之道

---

张熙

技术工程部 - 基础架构团队

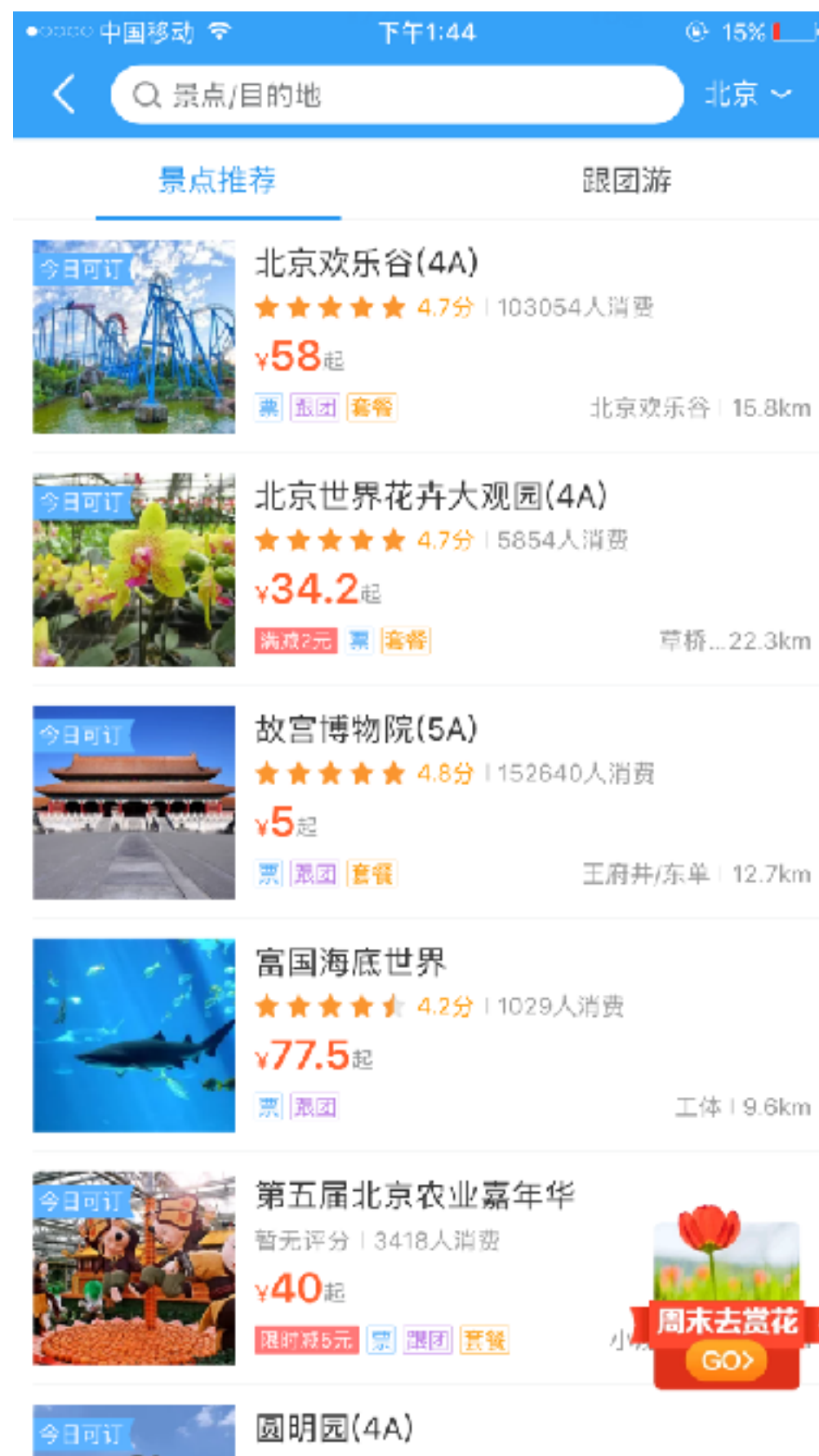
- 12年加入美团
- 技术工程部 - 基础架构中心服务治理团队负责人
  - OCTO：分布式服务通信框架及服务治理系统
  - HULK：容器集群管理及弹性伸缩平台
- 专注于面向服务架构、服务治理、大规模分布式系统、高性能通信框架、容器化、弹性调度等领域

- 系统容量概述
- 门票业务演进历程
- 系统容量优化之道

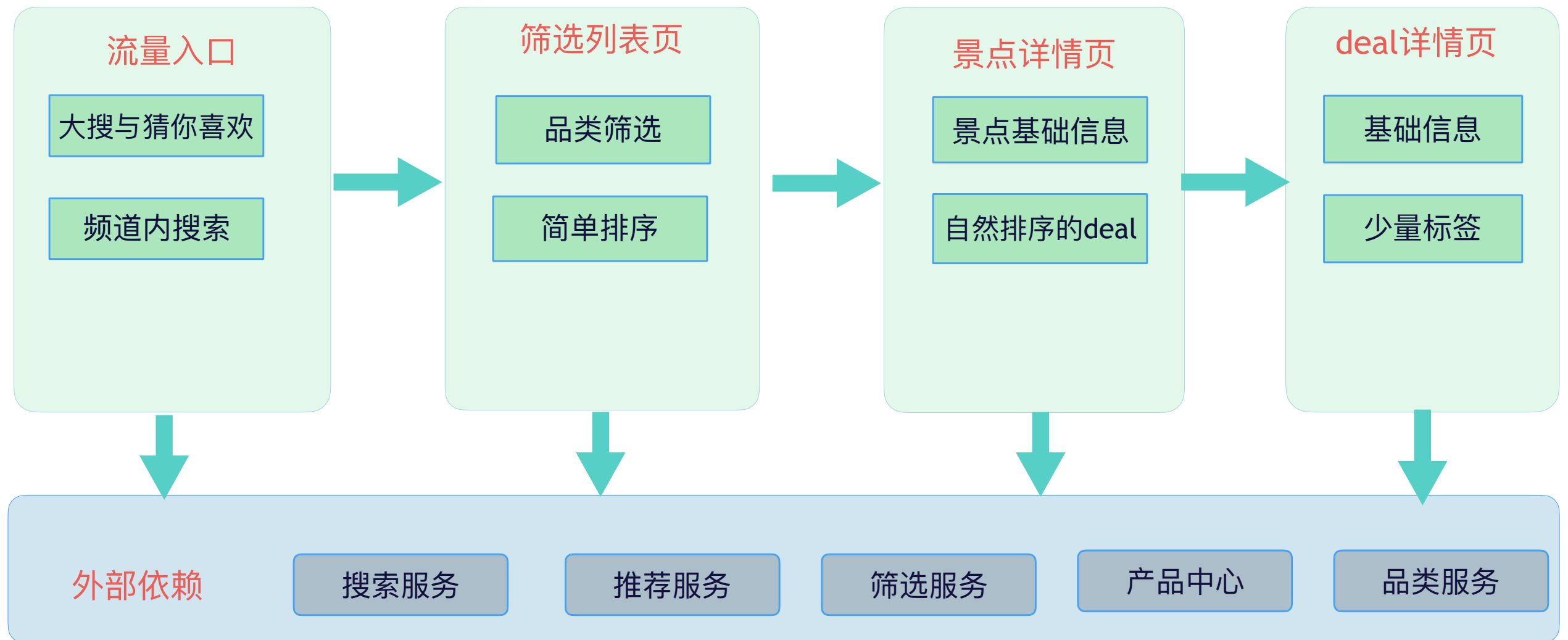
- 场景一：PM要做一个比较大的运营促销活动，老大过来问
  - 机器能抗住么？
  - 如果扛不住，需要加多少台机器？
- 场景二：被运维吐槽
  - 这么点调用量，还需要加机器？
  - CPU利用率怎么这么低？
- 场景三：经常收到调用方反馈
  - 你这个服务怎么一到高峰期就大量超时？

- 系统容量： **服务**在保障**SLA**情况下能承载的**极限**负荷
- 怎么做容量预估？
  - 量化：当前够不够，多少才满足要求
- 怎么诊断容量瓶颈？
  - 寻找短板：木桶效应
- 怎么做系统容量优化？
  - 提升效率：尽量少的资源

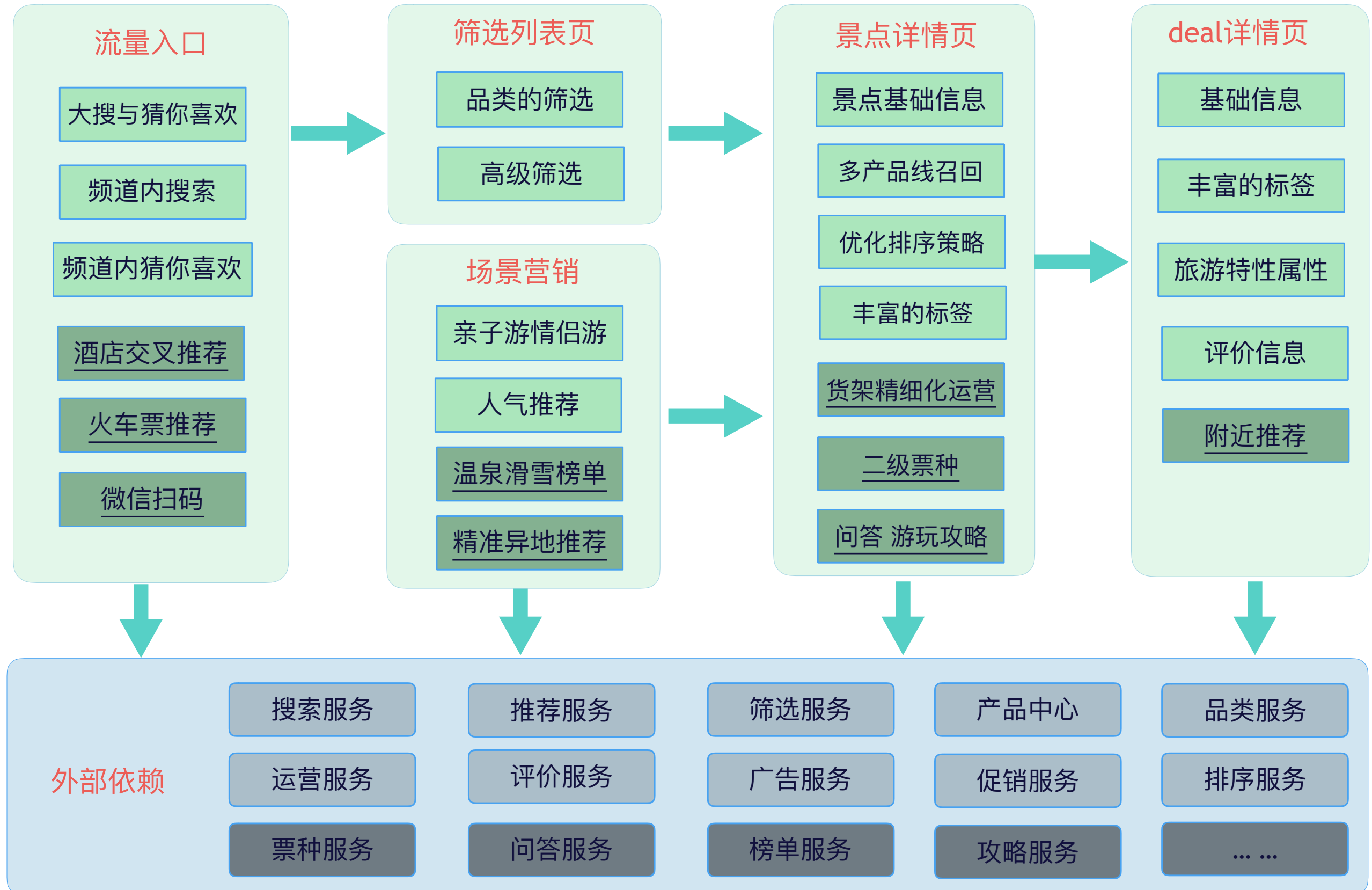
# 系统容量优化之道



## 门票业务逻辑 - V1版本



## 门票业务逻辑 - V2版本





- 业务背景
  - 快速增长
  - 产品逻辑复杂
  - 外部依赖增多
- 期间如何持续保障好系统容量？
  - 评测
  - 诊断
  - 优化
  - 规划

新纪录!

2016年10月2日

美团点评景点门票

单日销量突破 **100** 万张

单日销售额超过 **1** 亿元

创行业历史新高!

美团点评，国内最大的在线门票交易平台  
为全国超过2万个景区景点提供业界领先的专业票务解决方案

美团 美团 meituan.com 大众点评 dianping.com

— 酒旅事业群度假事业部 —

- 评测容量
  - 单机
    - 单台服务器压测的阈值
    - 关注指标：qps、RT50/90/99、load、gc
  - 集群
    - DB限制
    - Cache限制
    - 下游依赖限制
- 评测方式
  - 引流压测：受集群实际流量限制、需高峰期、无需清理数据
  - 模拟压测：不受实际流量限制、时间灵活、适合读请求
  - 全链路压测：最准确、需要底层组件支持

事务模式

递增模式

初始用户数10

递增用户数10

结束用户数40

\*QPS=并发用户数\*1000/响应时间(ms)

运行时长(s)60

单施压机最大并发数5

\*默认100,取值范围[4,100]

施压机数8

\*施压机数=(并发用户数-1)/单施压机最大并发数+1

压力工具所在机房大兴

\*线上压测选大兴，线下选永丰

- poi详情页攻略-杨前利 [查看](#)
- POIID获取POI简介-杨前利 [查看](#)

配比4

配比4

falcon监控

☐ 是否需要与高峰时段进行对比

分组名	主机名(Falcon endpoint)	falcon环境	监控阈值		
被压机器	dx-trip-volga09	0	cpu.idle	<	3
			load.1min	>	16
			jvm.fullgc.count	>	5

Console输出

测试报告

Falcon监控数据

Octo监控数据

对比报告

测试配置

基线构建编号 

260

指标	优化前		优化后	
	#260		#268	
并发数	50		100	↗
单施压机最大并发数	5		5	
QPS	247		586	↗
平均响应时间	77 ms		53 ms	↘
top90响应时间	145 ms		108 ms	↘
错误率	0.00%		0.00%	
请求量	44784		141441	↗

- 诊断瓶颈
  - 集群
    - DB：链接数、慢SQL、并发读写、主从同步、流量
    - Cache：超时率、热点key、命中率、流量
    - 下游依赖：超时率、容量、链路分析
  - 单机
    - CPU：top、jstack、JProfiler
    - 内存：jmap、MAT
    - 网络：netstat、iftop
    - 业务逻辑

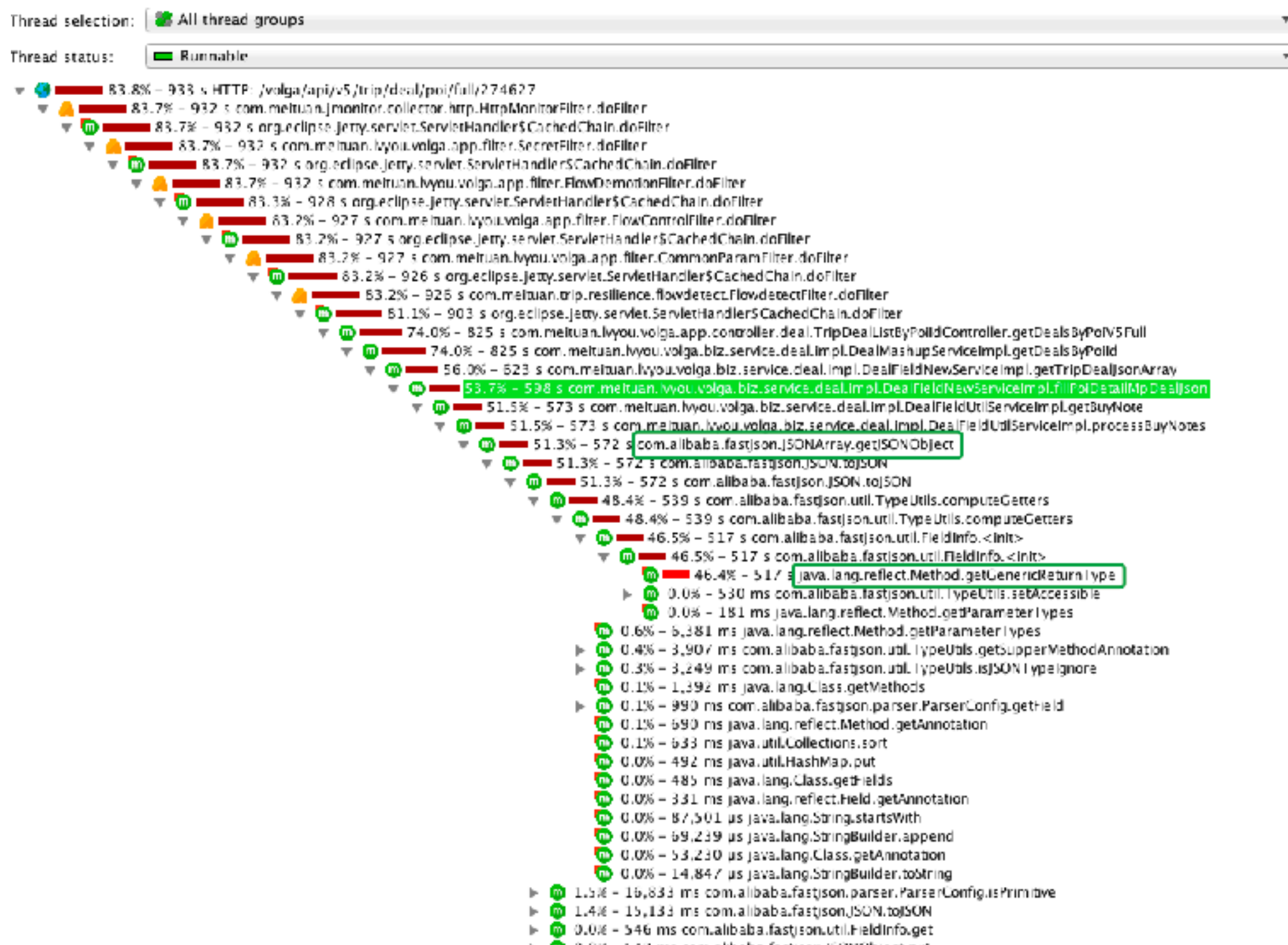
- 容量优化

- DB：读写分离、分库分表、SQL优化（Atlas、Zebra、SQLAdvisor）
- Cache：序列化、mget/mput、异步
- CPU：性能热点
- 下游依赖：链路分析、异步化
- 业务逻辑：精简、梳理

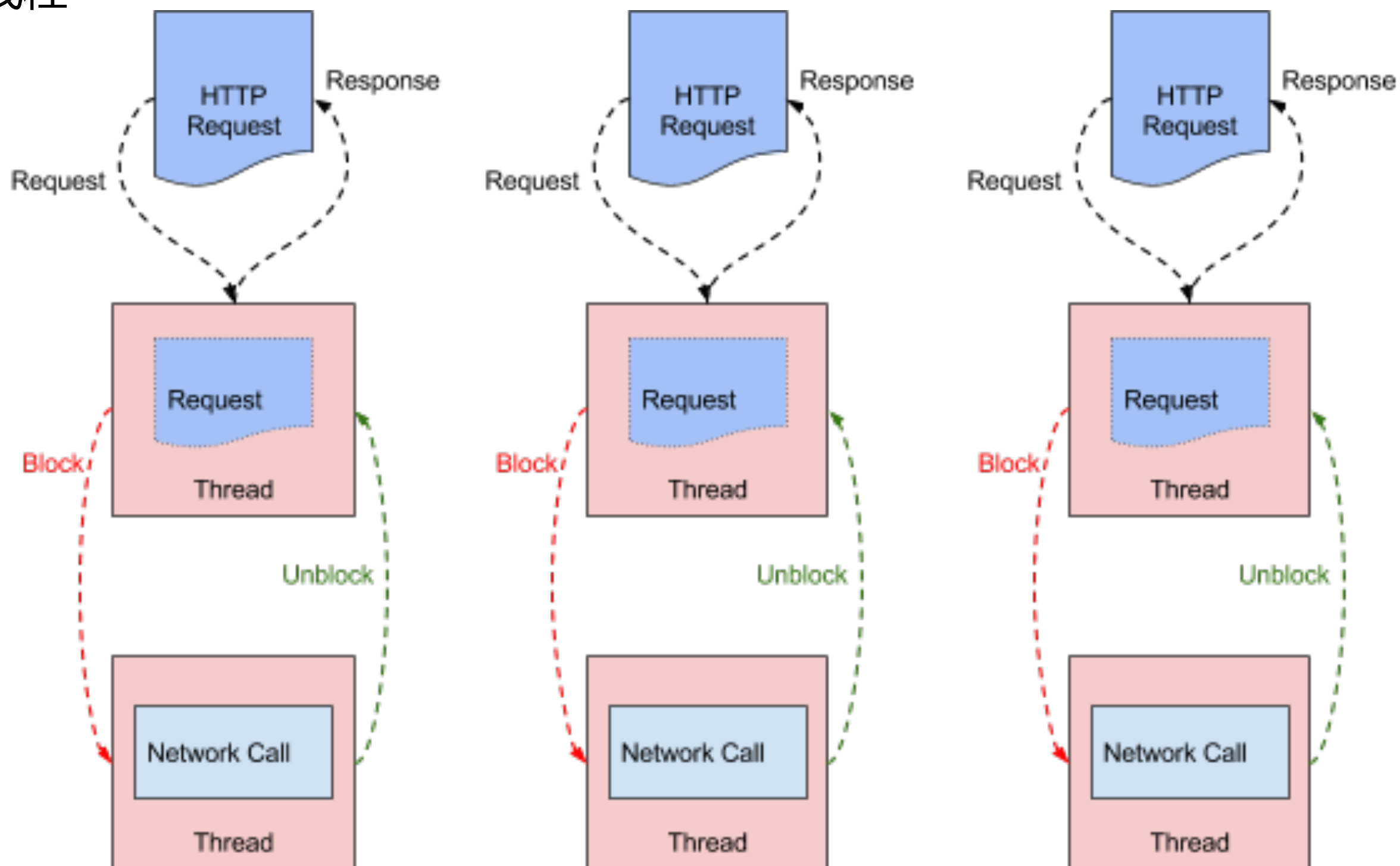


## 性能热点

- JSON解析
- 循环调用
- 签名/加密
- 日志输出

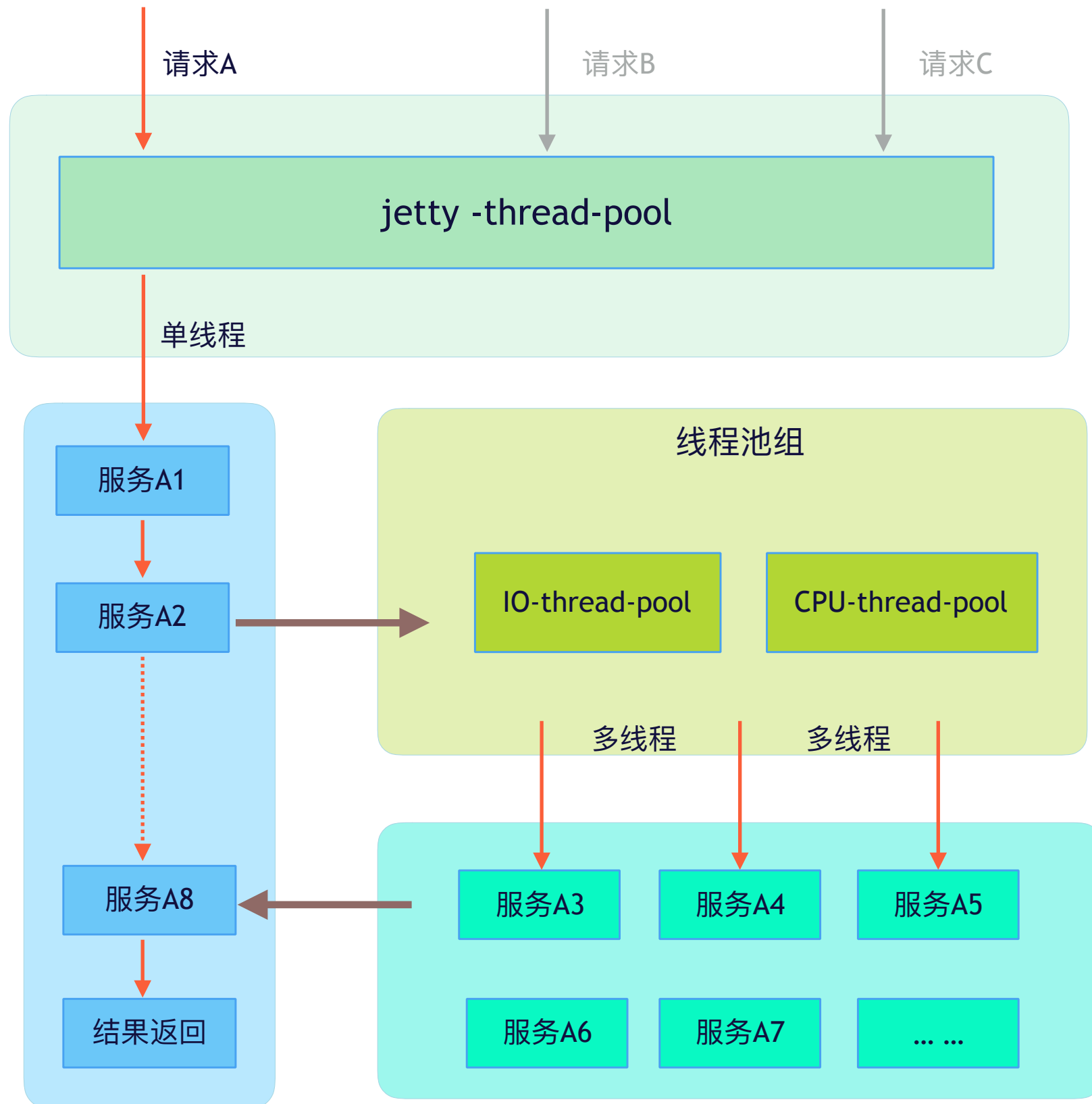


- 线程

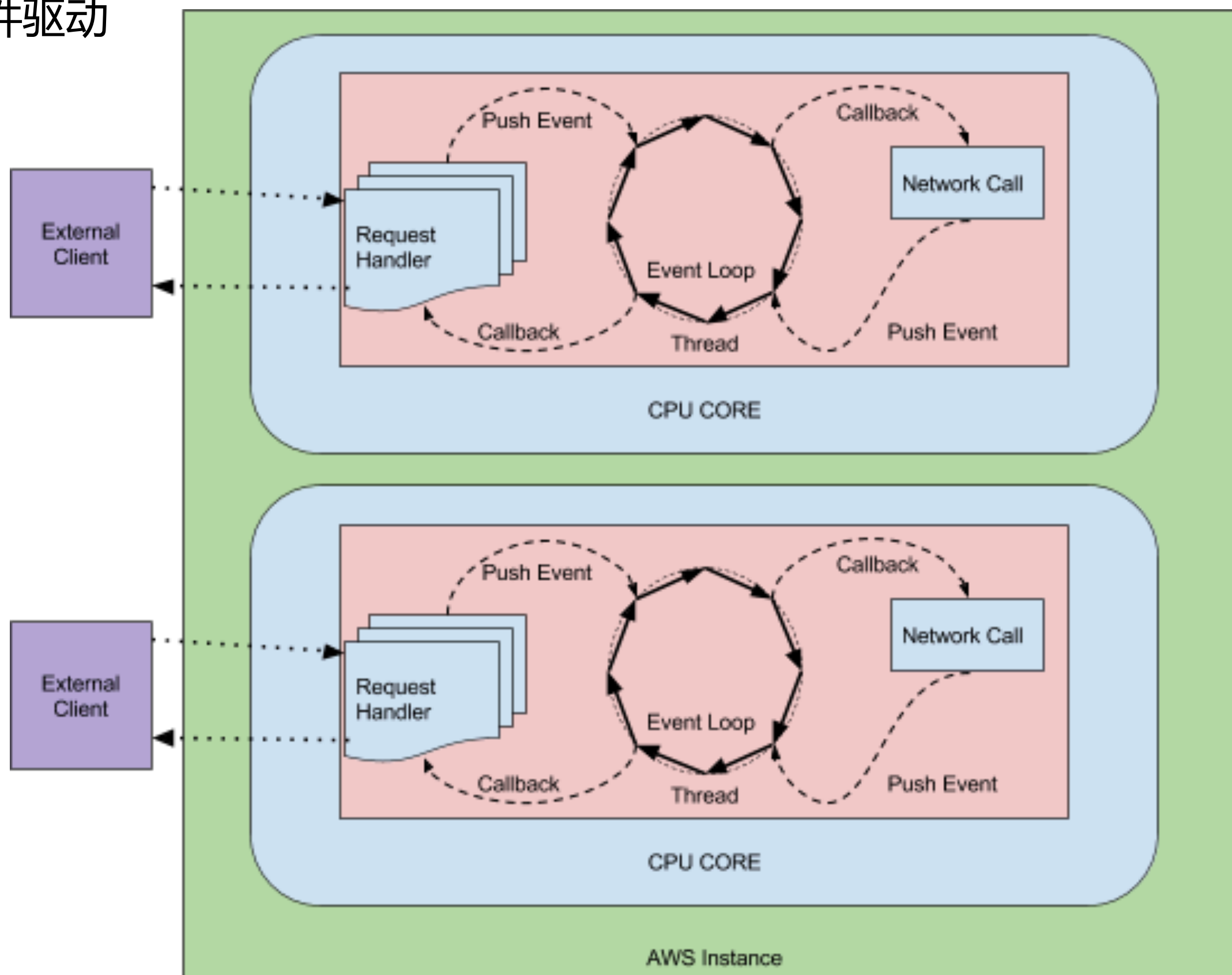




## • 线程池



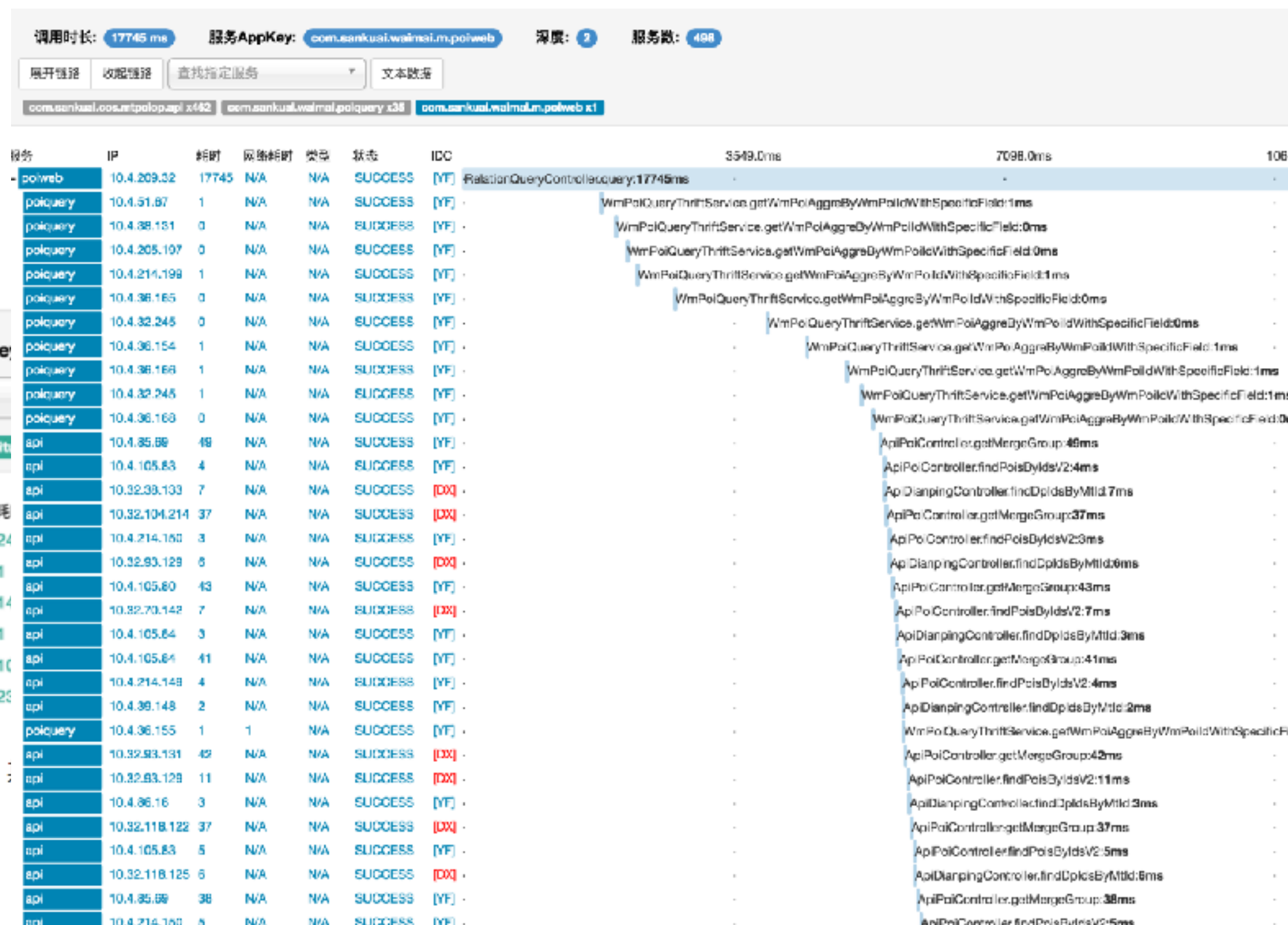
- 事件驱动



- 基础架构组件
  - RPC框架：mtthrift异步化
  - HTTP调用：Apache HttpAsyncClient
  - Cache：Cellar、Squirrel异步接口
- 异步化编程
  - Command模式
  - Java8 CompletableFuture
  - guava ListenableFuture

## 链路分析

- 链路过长
- 多次调用



如果一个api耗时242ms的接口，

服务	IP	耗时
groupapi	10.32.110.227	242
abtest	10.32.81.174	1
merge	10.32.106.192	14
filter	N/A	1
	N/A	10
sinai	10.32.57.121	25

- 多次调用
  - RPC：合并请求、批量接口
  - 缓存：mget、mput
  - DB：批量查询
- 业务逻辑梳理
  - 不必要的代码逻辑、远程调用
  - 冗余数据传输、精简数据

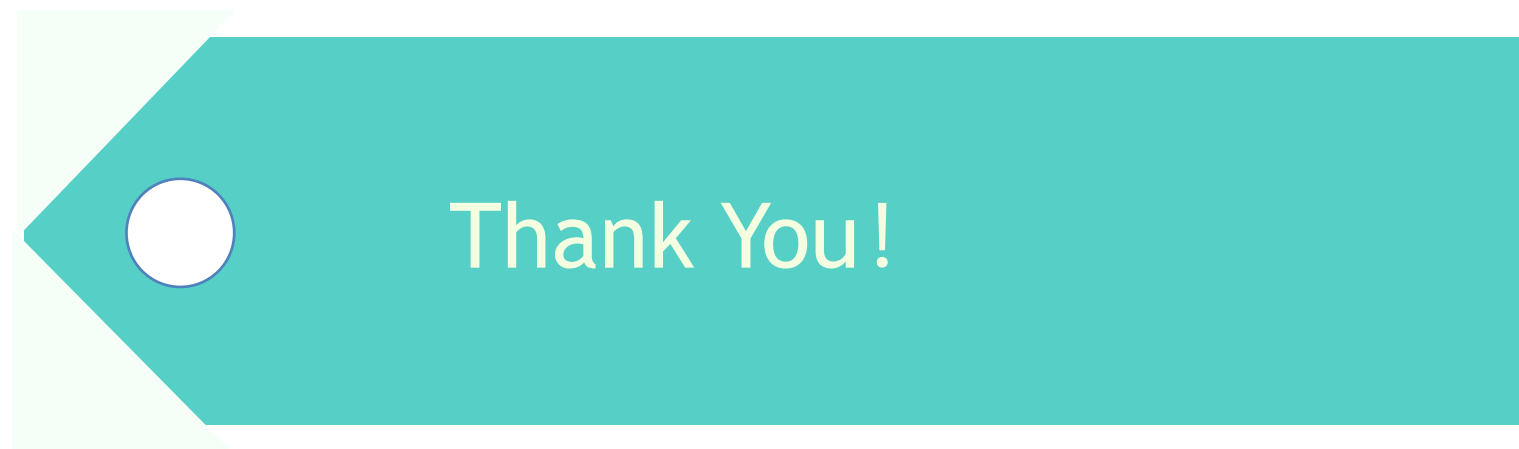
- 容量规划

- 单机容量 = 单台机器压测的阈值（满足SLA指标）
- 单机负荷 = 线上单台机器的峰值QPS
  - 峰值QPS估算 =  $(\text{日PV数} * 80\%) / (\text{每天秒数} * 20\%)$
- 集群容量 = 单机容量 \* 实际机器数 \* 容量系数
- 容量系数 = 取决于集群依赖资源的容量，如DB、Cache、下游服务
- 容量倍数 = 集群容量 / (实际机器数 \* 单机负荷)

- 机房容量

- 单机房水位：1.25倍容量、1.5倍容量、2倍容量
- 双机房水位：2倍容量、2.5倍、3倍容量

- 系统容量
  - 集群：机器数、DB、Cache、网络带宽
  - 单机：资源瓶颈、同步调用、业务逻辑
- 容量优化之道
  - 评测：引流、模拟、全链路
  - 诊断：DB、Cache、下游依赖、CPU、业务逻辑
  - 优化：性能热点、异步化、链路分析、逻辑梳理
  - 规划
    - $\text{QPS估算} = (\text{日PV数} * 80\%) / (\text{每天秒数} * 20\%)$
    - $\text{容量倍数} = \text{集群容量} / (\text{实际机器数} * \text{单机负荷})$



Email: [zhangxi@meituan.com](mailto:zhangxi@meituan.com)

微博: @lispmind